R With Git and Github

Aris Paschalidis

September 01, 2021

Section 1

Setup

Install Git

Git allows us to track changes to our documents (i.e. Git is version control).

- Check if git is already installed: `which git`
- Install git for Windows: https://gitforwindows.org/
- Install git for Mac: `brew install git`
- Install git for Linux:
  - Ubuntu or Debian: `sudo apt-get install git`
  - Fedora or RedHat: `sudo yum install git`

Configure Git

- Introduce yourself to git

```
usethis::use_git_config(user.name = "first last",
                        user.email = "github email")
```

- Ensure setup was successful

```
git config --global --list
```

Managing Git(Hub) Credentials[1]

- Adopt HTTPS
- Secure your account with 2FA
- Create a PAT: `usethis::create_github_token()`
- Store PAT into the Git credential store: `gitcreds::gitcreds_set()`

[1] Following the usethis guide

Section 2

Git Basics

Stage & Commit

- We can create and edit files within our project
- We can then **stage** and **commit** these changes

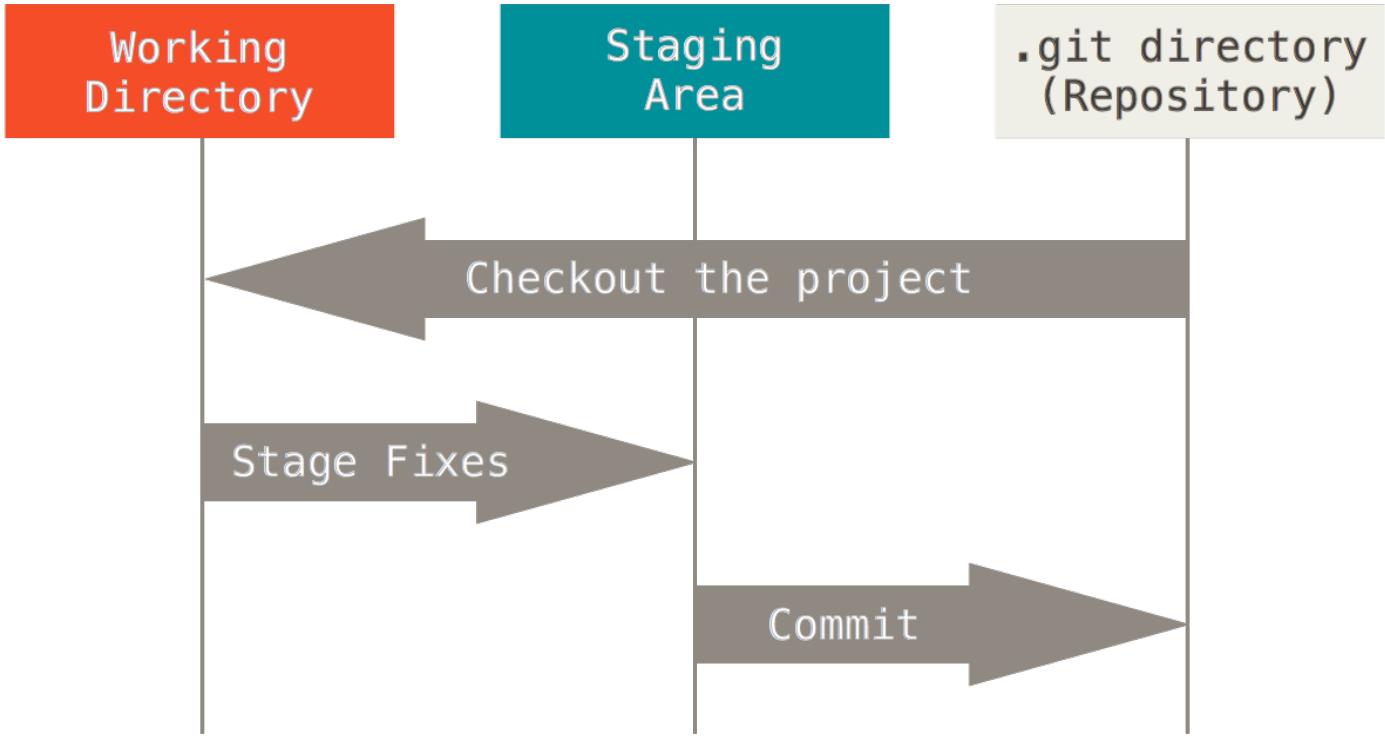

Figure 1: Git Workflow

Diff

- We can look at the set of *differences* between files to see what has changed

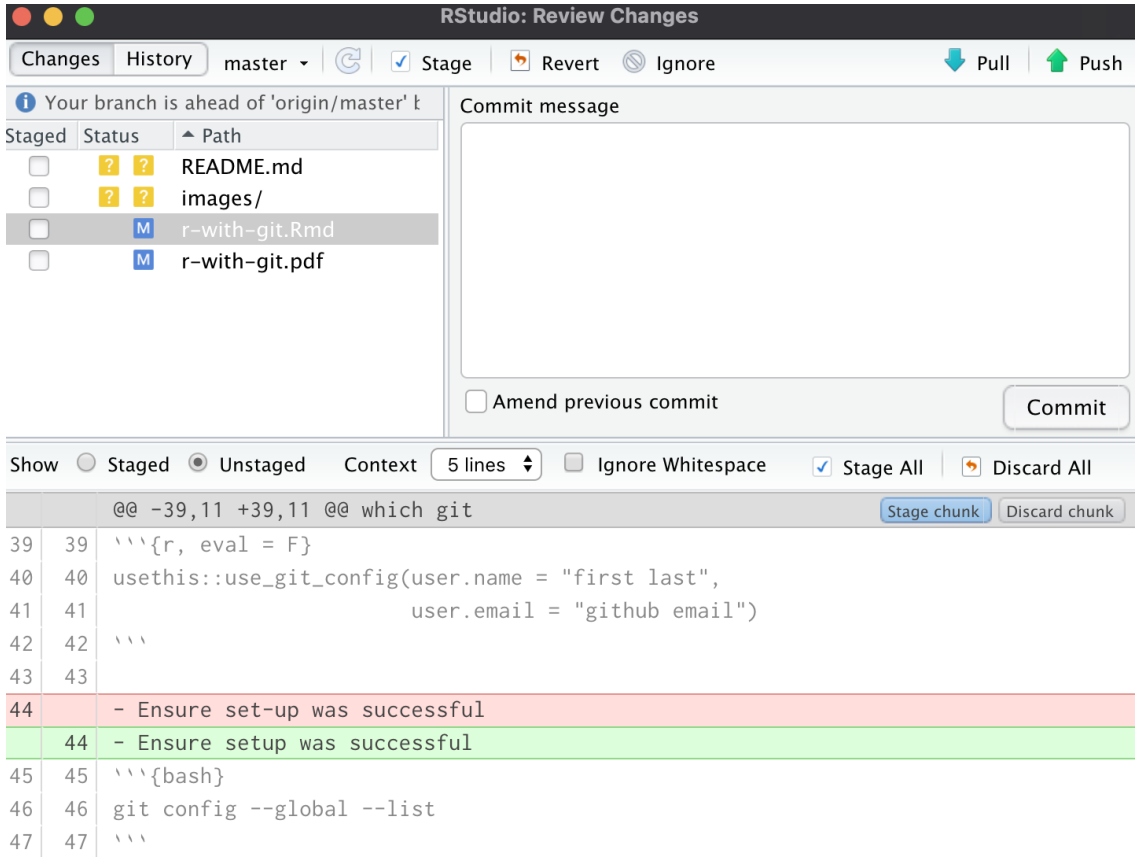

Figure 2: Example Diff

Commit Best Practices

Each commit should be:

- *Minimal:* A commit should only contain changes related to a single problem or feature
- *Complete:* A commit should add the functionality it claims to add
- *Concise, yet evocative:* At a glance, you should be able to understand what a commit does, but you should include enough detail so you can remember what was done

Time Travel

- Can look at old commits and access old code
- Can use your git client or Github
- Can revert back to a previous commit

Pull & Push Changes

- **Pulling** changes from Github allows us to access stored changes
- **Pushing** changes stores changes on Github

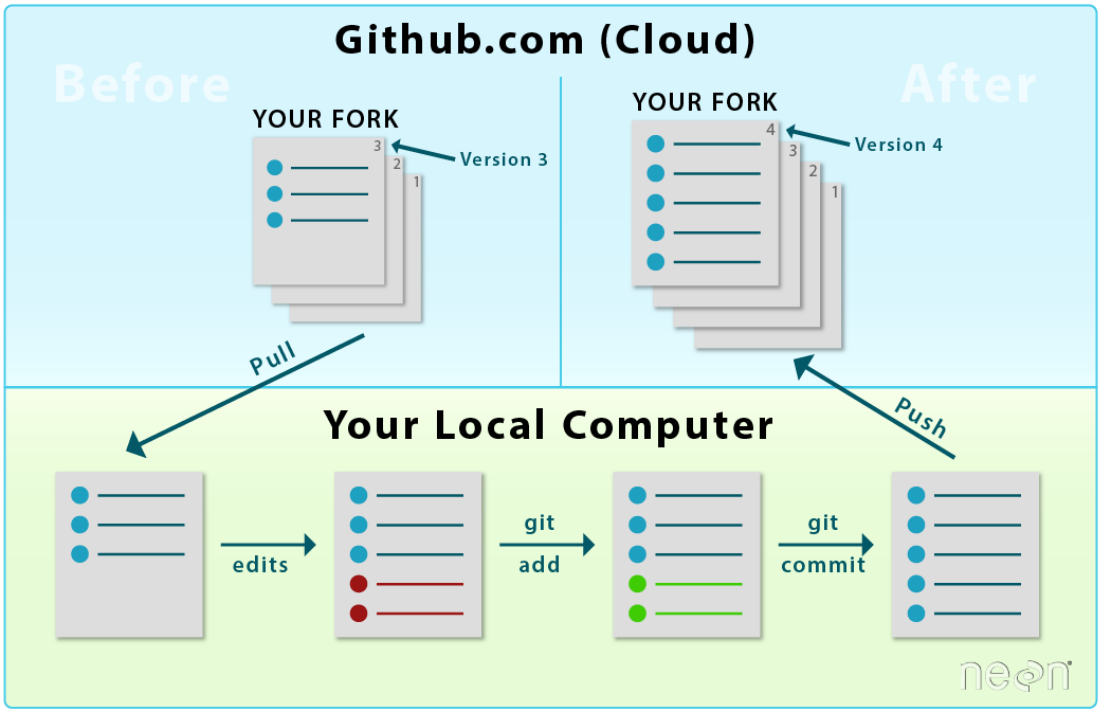

Figure 3: Accessing the Cloud

Section 3

Exercise: Use Git

Make a RStudio Project and Connect to Github

- Create a project: `usethis::create_project(path)`
- Use git: `usethis::use_git()`
- Connect a Github repo: `usethis::use_github()`

Commit & Push

- Make a .Rmd file and edit it
- Commit your change
- Make another change and commit it
- Push your changes

Section 4

Working With Others

Overview

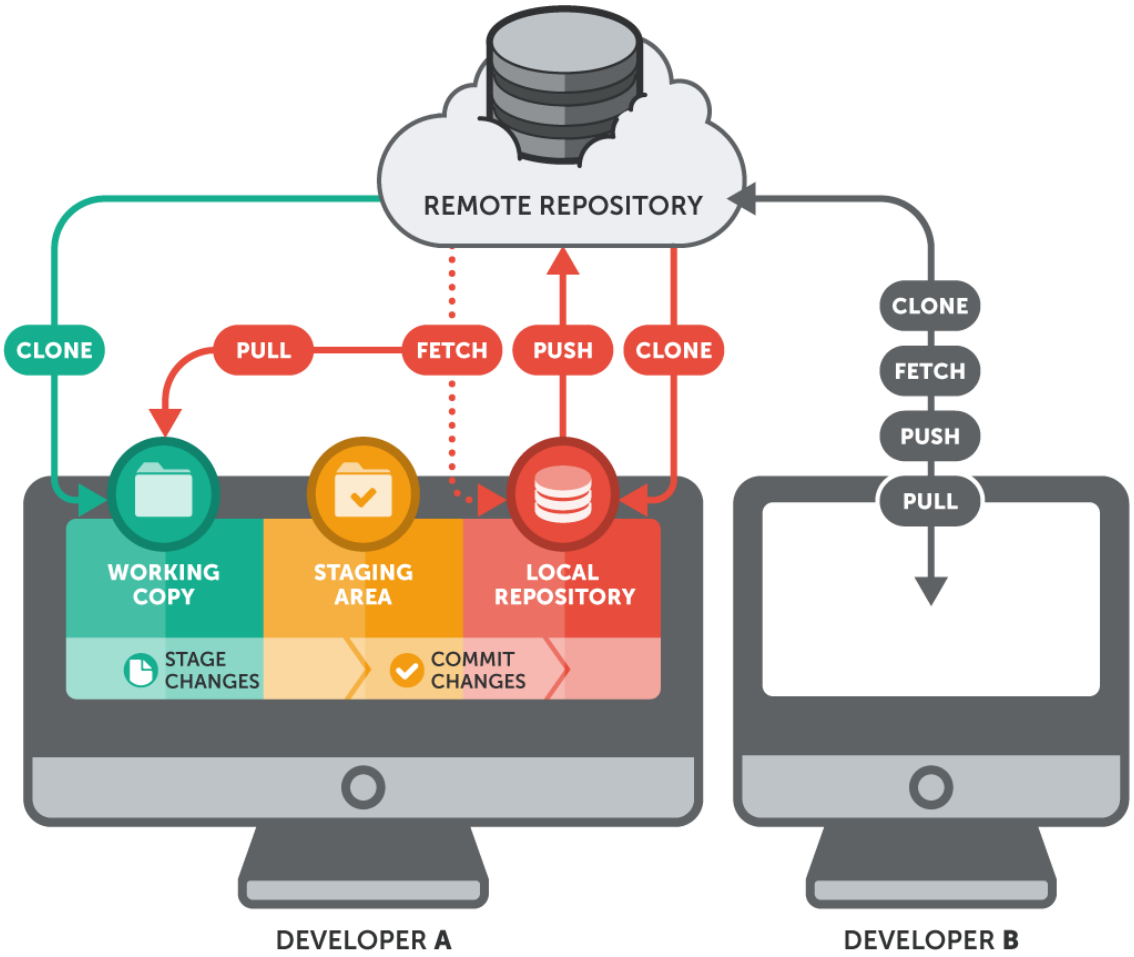

Figure 4: Multiple Developers

Fork & Clone

- Developer A makes a repo
- Developer B wants to make some changes
- Developed B **forks and clones** the repo:

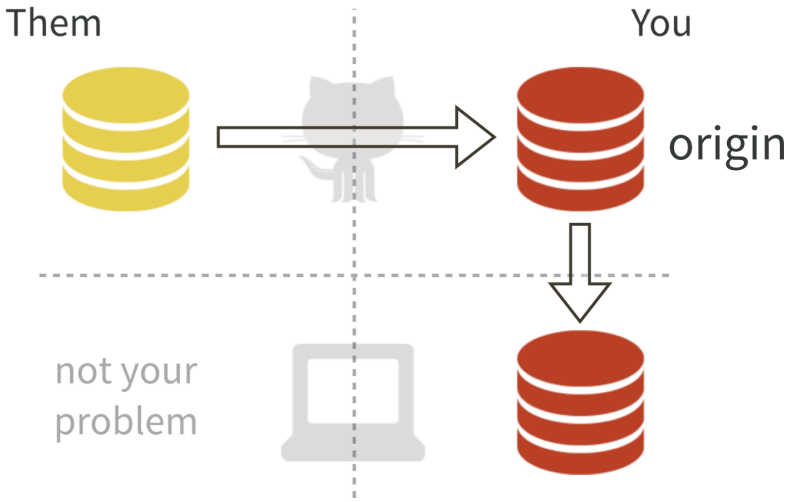

Figure 5: Fork and Clone

Branches

- Developer B makes a new **branch** to develop a new feature
- Allows Developer A to continue working on the main stream of development
- Developers A and B can work in parallel

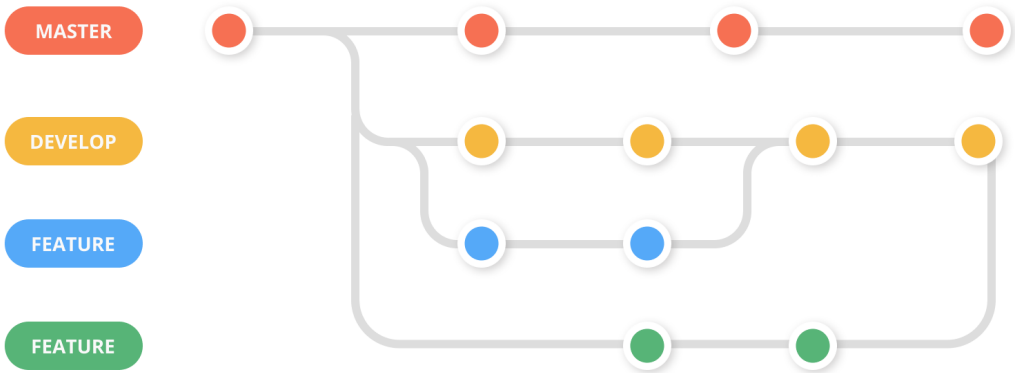

Figure 6: Branches

Pull Requests

- Developer B initiates a **pull request (PR)** asking Developer A to incorporate changes
- Developer A reviews the **PR**
- Developer A makes or requests changes
- Developer A merges the **PR**

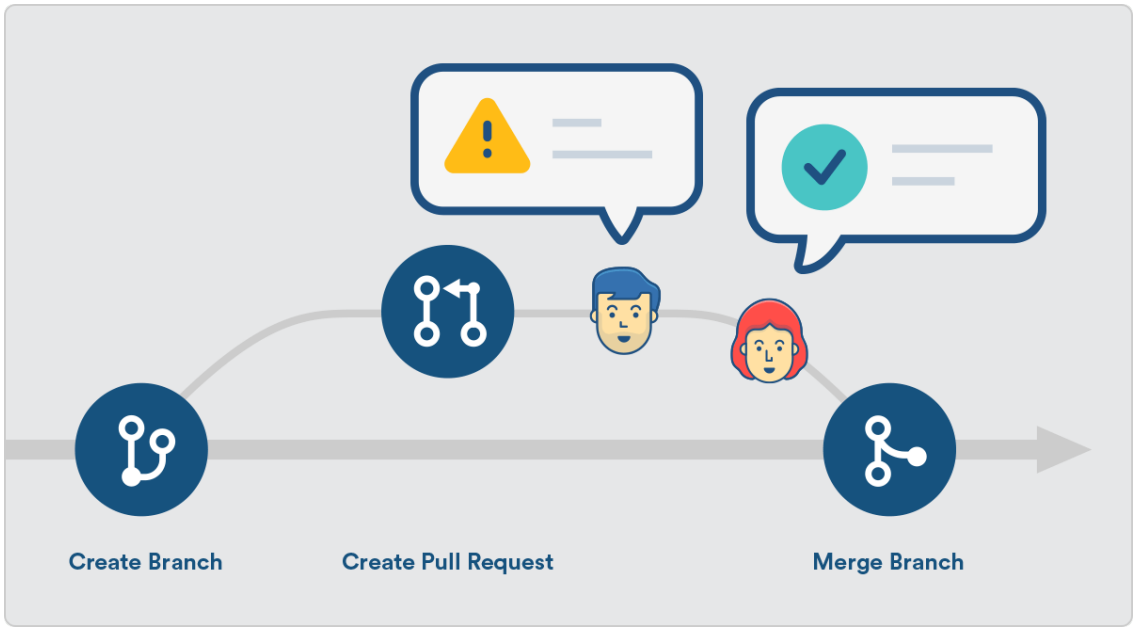

Figure 7: Pull Request Workflow

In Practice

Contributor:

- Fork and clone: `usethis::create_from_github("OWNER/REPO")`
- Create a branch: `usethis::pr_init(branch)`
- Push changes: `usethis::pr_push()`

Developer:

- Download a PR: `usethis::pr_fetch()`
- Push changes to branch: `usethis::pr_push()`
- Merge the PR on Github
- Delete the branch: `usethis::pr_finish()`

Section 5

Exercise: Create a Pull Request

Exercise: Create a Pull Request

- Fork and clone this repo:
  `usethis::create_from_github("arisp99/r-with-git")`
- Create a branch: `usethis::pr_init("USERNAME-fun-fact")`
- Add a geography fun fact and picture to the presentation!
- Push your changes: `usethis::pr_push()`

Section 6

Geography Fun Facts!

Greece

- The world's oldest weather station is at the base of the Acropolis

Figure 8: The Acropolis

Country

- Add your fun fact and image here!